UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/077,520	02/15/2002	Fintan Ryan	5181-78701	3344

58467   7590   03/19/2008

MHKKG/SUN
P.O. BOX 398
AUSTIN, TX 78767

EXAMINER
BOUTAH, ALINA A

ART UNIT	PAPER NUMBER
2143	

MAIL DATE	DELIVERY MODE
03/19/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES

Application Number: 10/077,520
Filing Date: February 15, 2002
Appellant(s): RYAN, FINTAN

Robert C. Kowert
Reg. No. 39,255
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed December 14, 2007 appealing from the Office action mailed August 23, 2007.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

6,785,706	Horman	8-2004
2002/0124061	Mossman	9-2002
2003/0014479	Shafron et al.	1-2003

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Claims 1-5, 7-18, 23-56, 60-66, 68, 69 and 72 are rejected under 35 U.S.C. 103(a) as being unpatentable over USPN 6,785,706 issued to Horman in view of US Pub. No. 2002/0124061 submitted by Mossman.

Regarding claim 1, Horman teaches a method for generating a batch configuration document for an intelligent device, the method comprising:

accessing a plurality of configuration files of the intelligent device (**abstract** and **col. 1, line 66-col. 2, line 6**: "the control server can change the *configurations* of the administered servers to a new desired *configuration* for each administered servers where the new

configuration is in synchronization with the *configuration* of each of said other administered

server, by generating synchronization instructions from parameterized synchronization *scripts*

stored in the control database" - in order to change the configuration, a configuration file must be

inherently accessed), wherein each of the one or more configuration files includes configuration

information for one of a plurality of software components of the intelligent device (*col. 5, lines*

18-21 – "the characteristics that many administered servers could share are the *end-user*

application that runs on them, and the database definition that supports the application);

generating the batch configuration document from the plurality of configuration files,

wherein the batch configuration document includes the configuration information for the

plurality of software components of the intelligent device (col. 5, lines 43-55);

wherein, after said generating, the batch configuration document is accessible for use in

configuring the plurality of software components of the intelligent device whose configuration

files were used in said generating the batch configuration document (col. 5, lines 43-55; lines 64

to col. 6, lines 5).

However, Horman does not explicitly teach the configuration files being accessed on the

intelligent device itself. Mossman teaches accessing configuration files on an intelligent device

(abstract; figure 5). At the time the invention was made, one of ordinary skill in the art would

have been motivated to access configuration files on an intelligent device in order to configure a

plurality of parameters of a target device, therefore optimizing the device for its intended use

[0005].

Regarding claim 2, Horman teaches the method as recited in claim 1, wherein said accessing the plurality of configuration files and said generating the batch configuration document are performed by executing a script on the intelligent device, wherein the script includes one or more executable instructions for selecting the plurality of configuration files to be accessed and one or more executable instructions for performing said generating the batch configuration document (abstract; col. 1, line 66 to col. 2, line 6; figure 1).

Regarding claim 3, Horman teaches the method as recited in claim 1, further comprising, prior to said accessing the plurality of configuration files, configuring the plurality of software components of the intelligent device, wherein said configuring the plurality of software components sets the configuration information in the plurality of configuration files (col. 1, line 66 to col. 2, line 6).

Regarding claim 4, Horman teaches the method as recited in claim 1, further comprising transferring the batch configuration document to another intelligent device for use in configuring one or more software components of the other intelligent device (col. 2, lines 28-41).

Regarding claim 5, Horman teaches the method as recited in claim 1, wherein the batch configuration document further includes configuration information for one or more software

components of one or more other intelligent devices (abstract; col. 1, line 66 to col. 2, line 6;

figure 1).

Regarding claim 7, Horman teaches the method as recited in claim 1, further comprising

configuring one or more of the plurality of software components of the intelligent device using

the batch configuration document, wherein said configuring comprises applying the

configuration information from the batch configuration document to one or more of the plurality

of configuration files, wherein each of the one or more of the plurality of configuration files is

associated with one of the one or more of the plurality of software components of the intelligent

device (col. 5, lines 43-55; lines 64 to col. 6, lines 5).

Regarding claim 8, Horman teaches the method as recited in claim 7, wherein said

configuring the one or more of the plurality of software components of the intelligent device

further comprises initializing each of the one or more of the plurality of software components,

wherein said initializing uses the configuration information from the one or more configuration

files associated with the particular component (col. 7, lines 32-36).

Regarding claim 9, Horman teaches the method as recited in claim 1, wherein the

plurality of software components includes software application programs (col. 3, lines 6-21).

Regarding claim 10, Horman teaches the method as recited in claim 1, wherein the plurality of software components includes system software components (col. 1, line 60 to col. 2, line 6).

Regarding claim 11, although Horman does not explicitly teach wherein the plurality of software components includes software drivers for hardware components, one of ordinary skill in the art would have recognize that this is well known in the art. In order for any application to work in a computer, there must be a hardware driver that enables the hardware to perform functions. One of ordinary skill in the art would have been motivated to include software drivers in order to enable the hardware to perform functions.

Regarding claim 12, Horman teaches the method as recited in claim 1, wherein at least one of the plurality of configuration files includes operating system configuration information for the intelligent device (col. 8, lines 14-16).

Regarding claim 13, Mossman teaches the method as recited in claim 1, wherein the batch configuration document is a markup language document [0091].

Regarding claim 14, Mossman teaches the method as recited in claim 13, wherein the markup language is eXtensible Markup Language (XML) [0091].

Regarding claim 15, Mossman teaches the method as recited in claim 1, wherein the batch configuration document and the plurality of configuration files conform to an eXtensible Markup Language (XML) Document Type Definition (DTD) [0091].

Claim 16 is an obvious variation of claim 1. Therefore claim 16 is rejected under the same rationale.

Regarding claim 17, Horman teaches the method as recited in claim 16, wherein said applying the configuration information from the batch configuration document to each of the one or more configuration files comprises replacing one or more current parameter values in the particular configuration file with new parameter values from the batch configuration document (col. 2, lines 50-65).

Regarding claim 18, Horman teaches the method as recited in claim 16, wherein said accessing and said applying are performed by executing a script on the intelligent device, wherein the script includes one or more executable instructions for accessing the batch configuration document and one or more executable instructions for selecting the one or more configuration files to be configured (col. 2, lines 50-65).

Claims 23-29 are similar to claims 9-15, respectively, therefore are also rejected under the same rationale.

Regarding claim 30, Mossman teaches the method as recited in claim 16, further comprising rebooting the intelligent device after said applying the configuration information from the batch configuration document to the one or more configuration files, wherein said rebooting applies the configuration information from the one or more configuration files to one or more of the plurality of software components of the intelligent device [0153].

Regarding claim 31, Mossman teaches the method as recited in claim 16, further comprising initializing one or more of the plurality of software components of the intelligent device after said applying the configuration information from the batch configuration document to the one or more configuration files, wherein, in said initializing, each of the one or more of the plurality of software components is initialized using the configuration information from each of the one or more configuration files associated with the particular component [0108].

Regarding claim 32, Horman teaches the method as recited in claim 16, further comprising generating the batch configuration document on a different intelligent device prior to said accessing (col. 8, lines 45-55).

Claim 33 is similar to claim 1, therefore is rejected under the same rationale.

Regarding claim 34, Homan teaches the method as recited in claim 33, wherein said configuring comprises applying configuration information from the batch configuration document generated on the first intelligent device to one or more configuration files on the second device, wherein each of the one or more configuration files on the second intelligent device is associated with one of the one or more software components of the second intelligent device (abstract; col. 1, line 66 to col. 2, line 6).

Claims 35-37 are similar to claims 13, 15, and 11, respectively, therefore are rejected under the same rationale.

Regarding claim 38, Homan teaches the method as recited in claim 33, wherein at least one of the plurality of configuration files on the first intelligent device includes operating system configuration information for the first intelligent device, wherein the batch configuration document includes the operating system configuration information, and wherein said configuring the one or more software components of the second intelligent device comprises configuring an operating system of the second intelligent device using the operating system configuration information of the first intelligent device from the batch configuration document (abstract; col. 1, line 66 to col. 2, line 6).

Regarding claim 39, Horman teaches the method as recited in claim 33, further

comprising rebooting the second intelligent device after said configuring, wherein said rebooting

applies the configuration information from the batch configuration document to the one or more

software components of the second intelligent device (abstract; col. 1, line 66 to col. 2, line 6).

Regarding claim 40, Horman teaches the method as recited in claim 33, further

comprising: storing the generated batch configuration document on a server, wherein the server

is coupled to the second intelligent device via a network (col. 1, line 60 to col. 2, line 6); and

downloading the stored batch configuration document to the second intelligent device (col. 6,

lines 6-17); wherein said configuring the one or more software components of the second

intelligent device uses the downloaded batch configuration document (col. 8, lines 45-51).

Claim 41 is similar to claim 1 therefore is rejected under the same rationale. However,

claim 41 further recites accessing on or more configuration files on each of the plurality of

intelligent devices and the batch configuration document is accessible for use in configuring the

plurality of intelligent devices whose configuration files were used in said generating the batch

configuration document. Horman teaches these limitations in the abstract, and col. 5, lines 43-55.

Regarding claim 42, Horman teaches the method as recited in claim 41, wherein the batch

configuration document is further accessible for use in configuring other pluralities of intelligent

devices (abstract).

Claims 43 and 44 are similar to claims 3 and 4, therefore are rejected under the same

rationale.

Claims 45-47 are similar to claims 12, 13, and 15, respectively, therefore are rejected

under the same rationale.

Claims 48-55 are similar to claims 41-47, therefore are rejected under the same rationale.

Regarding claim 56, Horman teaches an intelligent device, comprising:

a processor (inherent in figure 4);

a plurality of software components; a plurality of configuration files, wherein each of the

plurality of configuration files is associated with one of the plurality of software components,

and wherein each of the plurality of configuration files includes configuration information for its

associated component (abstract; col. 1, line 66 to col. 2, line 6); and

a memory operable to store program instructions, wherein the program instructions are

executable by the processor to:

open a batch configuration document, wherein the batch configuration document

comprises configuration information for the plurality of software components of the intelligent

device (figure 5A); and

apply the configuration information from the batch configuration document to the plurality of configuration files of the intelligent device (figure 5A).

However, Horman does not explicitly teach the configuration files being accessed on the intelligent device itself. Mossman teaches accessing configuration files on an intelligent device (abstract; figure 5). At the time the invention was made, one of ordinary skill in the art would have been motivated to access configuration files on an intelligent device in order to configure a plurality of parameters of a target device, therefore optimizing the device for its intended use [0005].

Claims 60-65 are similar to claims 44, 45, 46, 47, 30, and 31, respectively, therefore are rejected under the same rationale.

Claims 66, 68 and 69 are similar to claims 1, 16 and 15, respectively, therefore are rejected under the same rationale.

Claims 70 and 72 are similar to claims 41 and 47, respectively, therefore are rejected under the same rationale.

Claims 6, 19-22, 57-59, 67 and 71 are rejected under 35 U.S.C. 103(a) as being

unpatentable over Horman in view of Mossman, in further view of US Pub No. 2003/0014479

submitted by Shafron et al. (hereinafter Shafron).

Regarding claims 6 and 19, Horman-Mossman combination fails to explicitly teach

generating the batch configuration document comprises generating a Document Object Model

(DOM) tree from each of the accessed configuration files, wherein the configuration information

incorporated in the configuration document is accessed from the DOM trees generated from the

plurality of configuration files. Shafron teaches generating the batch configuration document

comprises generating a Document Object Model (DOM) tree from each of the accessed

configuration files, wherein the configuration information incorporated in the configuration

document is accessed from the DOM trees generated from the plurality of configuration files

[0005; 0032; 0052]. At the time the invention was made, one of ordinary skill in the art would

have been motivated to generate a DOM tree because DOM allows programs and scripts to

access and update the content, structure, and style of documents dynamically.

Regarding claim 20, Horman-Mossman fails to teach the method as recited in claim 19.

In an analogous art, Shafron teaches wherein said applying the configuration information of the

one or more configuration files comprises accessing the configuration information from the

DOM tree generated from the batch configuration document [0005; 0032; 0051]. At the time the

invention was made, one of ordinary skill in the art would have been motivated to access the

configuration information from the DOM tree generated from the batch configuration document

in because DOM allows programs and scripts to access and update the content, structure, and

style of documents dynamically.

Regarding claim 21, Horman-Mossman fails to teach the method as recited in claim 16.

In an analogous art, Shafron teaches wherein said accessing the batch configuration document

comprises generating a Document Object Model (DOM) tree for each of the one or more

configuration files from the configuration information in the batch configuration document,

wherein each of the generated DOM trees comprises the configuration information for its

associated configuration file [0005; 0032; 0051]. At the time the invention was made, one of

ordinary skill in the art would have been motivated to generate a DOM tree because DOM allows

programs and scripts to access and update the content, structure, and style of documents

dynamically.

Regarding claim 22, Horman-Mossman fails to teach the method as recited in claim 21.

Shafron teaches wherein said applying the configuration information comprises: for each of one

or more of the plurality of software components of the intelligent device: calling a module

associated with the component; passing a DOM tree generated from one of the one or more

configuration files to the called module, wherein the configuration file is associated with the

component, and wherein the DOM tree includes configuration information for the component;

and the called module applying the configuration information from the DOM tree to the

configuration file associated with the component [0005; 0032; 0051]. At the time the invention

was made, one of ordinary skill in the art would have been motivated to generate a DOM tree

because DOM allows programs and scripts to access and update the content, structure, and style

of documents dynamically.

Regarding claim 57, 67 and 71, Horman-Mossman fails to teach the intelligent device as

recited in claim 56. Shafron teaches wherein the program instructions are further executable by

the processor to: generate a Document Object Model (DOM) tree from the batch configuration

document, wherein the DOM tree includes the configuration information for the plurality of

configuration files; and wherein, in said applying the configuration information to the plurality of

configuration files, the program instructions are further executable by the processor to access the

configuration information from the DOM tree generated from the batch configuration document

[0005; 0032; 0051]. At the time the invention was made, one of ordinary skill in the art would

have been motivated to generate a DOM tree because DOM allows programs and scripts to

access and update the content, structure, and style of documents dynamically.

Regarding claim 58, Horman-Mossman fails to teach the intelligent device as recited in

claim 56. Shafron teaches wherein, in said accessing the batch configuration document, the

program instructions are further executable by the processor to: generate a Document Object

Model (DOM) tree for each of the plurality of configuration files from the configuration

information in the batch configuration document, wherein each of the generated DOM trees

comprises the configuration information for its associated configuration file; wherein the

intelligent device further comprises a plurality of executable modules each associated with one

of the plurality of software components, wherein each of the plurality of executable modules is operable to apply configuration information to the particular one of the plurality of configuration files associated with the component associated with the executable module [0005; 0032; 0051]. At the time the invention was made, one of ordinary skill in the art would have been motivated to generate a DOM tree because DOM allows programs and scripts to access and update the content, structure, and style of documents dynamically.

Regarding claim 59, Horman-Mossman fails to teach the intelligent device as recited in claim 56. Shafron teaches wherein in said applying the configuration information to the plurality of configuration files, the program instructions are further executable by the processor to: for each of the plurality of software components of the intelligent device: call one of the plurality of modules, wherein the called module is associated with the component; and pass a DOM tree generated from one of the plurality of configuration files to the called module, wherein the configuration file is associated with the component, and wherein the DOM tree includes configuration information for the component; and wherein the called module is operable to apply the configuration information from the DOM tree to the configuration file associated with the component [0005; 0032; 0051]. At the time the invention was made, one of ordinary skill in the art would have been motivated to generate a DOM tree because DOM allows programs and scripts to access and update the content, structure, and style of documents dynamically.

(10) Response to Argument

Appellant's arguments have been fully considered but they are not persuasive.

It should be noted that Appellant employs broad language, which includes the use of word, and phrases (i.e. configuration information), which have broad meanings in the art. In addition, Appellant has not argued any narrower interpretation of the claim language, nor amended the claims significantly enough to construe a narrower meaning to the limitations. As the claims breadth allows multiple interpretations and meanings, which are broader than Applicant's disclosure, the Examiner is forced to interpret the claim limitations as broadly and as reasonably possible, in determining patentability of the disclosed invention. Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See In re Van Geuns, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir.1993). Failure for Appellant to significantly narrow definition/scope of the claims and supply arguments commensurate in scope with the claims implies the Appellant intends broad interpretation be given to the claims. The Examiner has interpreted the claims with scope parallel to the Appellant in the response.

In response to Appellant's argument that Horman in view of Mossman fails to teach "accessing a plurality of configuration files on the intelligent device, wherein each of the one or more configuration files includes configuration information for one of a plurality of software components of the intelligent device," the PTO respectfully disagrees and submits that this is being taught by the cited reference.

Horman teaches the configuration file including configuration information for one or a plurality of software component. Horman teaches a control server that changes the configuration of administered servers (abstract). The configuration of the administered server is interpreted as

the configuration file of the intelligent device as claimed. Col. 5, lines 22-32, for example,

define what is in the configuration file. In this case, it is referred to the ***end-user application***. It

is well known in the computing art that an end-user application comprises software components.

In response to Appellant's argument that Mossman does not overcome the deficiency of

Horman in teaching "accessing a plurality of configuration files on the intelligent device," the

PTO respectfully disagrees and respectfully submits that Mossman does teach this limitation. As

admitted by Appellant, data is collected from the user and stored on the server. In order to

collect the data from the user, it must be accessible to the server.

In response to Appellant's argument that Horman in view of Mossman fails to teach

"generating a batch configuration documents from the plurality of configuration files, wherein

the batch configuration document includes the configuration information for the plurality of

software components of the intelligent device," the PTO respectfully disagrees and submits that

this is being taught by Horman. Col. 5, lines 43-55, for example, teaches generating a batch file

that is specific to a group. The batch file includes the end-user application as well as database

definition. As explained above, the end-user application comprises software component, as well

known in the art.

In response to Appellant's argument that Horman in view of Mossman fails to teach "the

batch configuration document is accessible for use in configuring a plurality of software

components of an intelligent device whose configuration files were used in generating the batch

configuration document," as discussed above, Horman teaches a group batch that includes

components of an intelligent device. The control server synchronizes each of the group batches

by the creation of parameterized synchronization scripts. In other words, when the synchronization script is executed, the software component of the intelligent device is changed.

In response to Appellant's argument that there is no suggestion to combine the references, the examiner recognizes that obviousness can only be established by combining or modifying the teachings of the prior art to produce the claimed invention where there is some teaching, suggestion, or motivation to do so found either in the references themselves or in the knowledge generally available to one of ordinary skill in the art. See *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988) and *In re Jones*, 958 F.2d 347, 21 USPQ2d 1941 (Fed. Cir. 1992).

In view of Supreme Court Decision in KRS International Co. v. Teleflex Inc., 550 U.S. -, 82 USPQ2d 11385 (2007), the Supreme Court stated that the Federal Circuit erred when it applied the well-known teaching-suggestion-motivation (TSM) test in an overly rigid and formalistic way. According to the Supreme Court, the TSM test is one of a number of valid rationales that could be used to determine obviousness. It is *not* the only rationale that may be relied upon to support a conclusion of obviousness.

The cited arts support the differences between claim 1 and 41. See above rejection and above explanation.

The scope of claim 16 is an obvious variation of claim 1. Although the wordings are different, these claims essentially claim the same inventive entity.

Claim 1 recites a method for generating a batch configuration document for an intelligent device, the method comprising:

accessing a plurality of configuration files of the intelligent device, wherein each

of the one or more configuration files includes configuration information for one of a

plurality of software components of the intelligent device; and

generating the batch configuration document from the plurality of configuration

files, wherein the batch configuration document includes the configuration information

for the plurality of software components of the intelligent device;

wherein, after said generating, the **batch configuration document is accessible**

for use in configuring the plurality of software components of the intelligent device

whose configuration files were used in said generating the batch configuration document.

Appellant argues that claim 16 is different from claim 1 because it recites "a method for

configuring a plurality of software components of an intelligent device. It includes accessing a

batch configuration document, wherein the batch configuration document comprises

configuration information for the polarity of software components of the intelligent device."

Firstly, it is noted that the difference in the wording in the preamble is not considered. A

preamble is generally not accorded any patentable weight where it merely recites the purpose of

a process or the intended use of a structure, and where the body of the claim does not depend on

the preamble for completeness but, instead, the process steps or structural limitations are able to

stand alone. See *In re Hirao*, 535 F.2d 67, 190 USPQ 15 (CCPA 1976) and *Kropa v. Robie*, 187

F.2d 150, 152, 88 USPQ 478, 481 (CCPA 1951).

Secondly, claim 16 recites the step of "accessing a batch configuration document." In

contrast, claim 1 recites that "the batch configuration document is accessible for use in

configuring the plurality of software components." Although the wordings are different, there is

no clear distinction between the limitations. Horman teaches this limitation as cited above in

claim 1.

Regarding claim 33, Horman in view of Mossman teaches generating a batch

configuration document from a plurality of configuration files on a first intelligent device and

configuring one or more software components of a second intelligent device using the batch

configuration document generated on the first intelligent device. As explained above, the control

server creates a synchronization script. The synchronization script is used to configure the

administered servers.

Regarding claims 56 and 60, Horman teaches a plurality of software components and a

plurality of configuration files, wherein each of the plurality of configuration files is associated

with one of the plurality of software components and wherein each of the plurality of

configuration files includes configuration information for its associated component. As explained

above, Horman teaches user-end applications (software components) and script files

(configuration files).

Regarding claim 2, Horman teaches wherein said accessing the plurality of configuration

files and said generating the batch configuration document are performed by executing a script

on the intelligent device, wherein the script includes one or more executable instructions for

selecting the plurality of configuration files to be accessed and one or more executable

instructions for performing said generating the batch configuration document (abstract; col. 1,

line 66 to col. 2, line 6; figure 1).

Regarding claim 3, Horman prior to said accessing the plurality of configuration files, configuring the plurality of software components of the intelligent device, wherein said configuring the plurality of software components sets the configuration information in the plurality of configuration files (col. 1, line 66 to col. 2, line 6).

Regarding claim 4, transferring the batch configuration document to another intelligent device for use in configuring one or more software components of the other intelligent device (col. 2, lines 28-41).

Regarding claim 5, Horman teaches the method as recited in claim 1, wherein the batch configuration document further includes configuration information for one or more software components of one or more other intelligent devices (abstract; col. 1, line 66 to col. 2, line 6; figure 1).Regarding claim 2, Horman teaches

Regarding claim 7, Horman teaches the method as recited in claim 1, further comprising configuring one or more of the plurality of software components of the intelligent device using the batch configuration document, wherein said configuring comprises applying the configuration information from the batch configuration document to one or more of the plurality of configuration files, wherein each of the one or more of the plurality of configuration files is associated with one of the one or more of the plurality of software components of the intelligent device (col. 5, lines 43-55; lines 64 to col. 6, lines 5).

Regarding claim 48, although it is not identical to claim 1, it is known in the art that any intelligent device, such as a computer, comprises a processor, memory, software, etc. Although not explicitly taught, these features are inherent in Horman's administered servers.

Regarding claim 8, it is well known in the art that in order to execute software, it must be initialized.

Regarding claim 11, although Horman does not explicitly teach wherein the plurality of software components includes software drivers for hardware components, one of ordinary skill in the art would have recognize that this is well known in the art. In order for any application to work in a computer, there must be a hardware driver that enables the hardware to perform functions. One of ordinary skill in the art would have been motivated to include software drivers in order to enable the hardware to perform functions.

Regarding claim 12, Horman teaches wherein at least one of the plurality of configuration files includes operating system configuration information for the intelligent device (col. 8, lines 14-16).

Regarding claim 13, Mossman teaches configuration files being in XML format. The use of batch files have been known in the art. It being a mark up language is a matter of design choice. Therefore, although not explicitly taught, this feature is obvious in the art.

Regarding claim 30, it is well known in the computer art that in order for any change or configuration to take place, the computer must be rebooted.

Regarding claim 6, by definition, DOM a platform and language-neutral interface that will allow programs and scripts to dynamically access and update the content, structure and style documents. The combined references teach scripts that update content of software on intelligent devices. As suggested by Shafron, one of ordinary skill in the art would employ a DOM as a matter of design choice [0032]. This argument also applies to the rejection of claims 67 and 71.

Regarding claim 15, Mossman wherein the batch configuration document and the plurality of configuration files conform to an eXtensible Markup Language (XML) Document Type Definition (DTD) [0091].

Regarding claim 17, Horman wherein said applying the configuration information from the batch configuration document to each of the one or more configuration files comprises replacing one or more current parameter values in the particular configuration file with new parameter values from the batch configuration document (col. 2, lines 50-65).

Regarding claim 18, Horman teaches wherein said accessing and said applying are performed by executing a script on the intelligent device, wherein the script includes one or more executable instructions for accessing the batch configuration document and one or more executable instructions for selecting the one or more configuration files to be configured (col. 2, lines 50-65).

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

/A. N. B./

Examiner, Art Unit 2143

Conferees

/Nathan J. Flynn/

Supervisory Patent Examiner, Art Unit 2154

/John Follansbee/

Supervisory Patent Examiner, Art Unit 2151